Information and Management Sciences

(NASA-CR-129825) ALTERNATIVES IN THE COMPLEMENT AND STRUCTURE OF NASA TELEPROCESSING RESOURCES (Auerbach Associates, Inc., Arlington, Va.) 30 Aug. 1972 74 p

N73-14188

CSCL 09B G3/08

Unclas 16408

AUERBACH®

ALTERNATIVES IN THE COMPLEMENT AND STRUCTURE OF NASA TELEPROCESSING RESOURCES

TECHNICAL REPORT
1958-100-TR-004

Submitted to:

NASA Headquarters
Under
Contract No. NASW-2285

August 30, 1972

AUERBACH Associates, Inc.
1501 Wilson Boulevard
Arlington, Virginia
22209

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

SECTION 1. INTRODUCTION

1.1 PURPOSE

The purpose of this report is to convey the results of work under Contract NASW-2285 to the NASA Headquarters Office of Tracking and Data Acquisition. The major task of this contract, to "identify and recommend alternatives in the complement and structure of NASA teleprocessing resources - 1975 through 1985," is the source of most of the material presented here and of the three Appendices, in toto.

The purpose of the contract was to identify technical innovations which would have an impact on NASA data processing and describe as fully as possible the development work necessary to exploit them. Seven of these options for NASA development, as the opportunities to participate in and enhance the advancing information systems technology were called, are reported here. The appendices contain a detailed treatment of three of the options, involving minicomputers, mass storage devices and software development techniques. These three areas were picked by NASA as having the most potential for improving their operations.

1.2 SCOPE

The contract called for approximately two manyears over a period of one year. Hence, an exhaustive survey of NASA information processing requirements was not attempted, although a limited survey was completed. The options reported, particularly the three selected by NASA for elaboration, owe more to a knowledge of data processing technology than to familiarity with NASA operations. A thorough survey of technical developments, current and anticipated, was conducted for this purpose.

It is not intended that results of the three developments outlined in the appendices be limited to tracking and data acquisition applications (the sponsor's field). For example, members of the standardized minicomputer family, if developed as outlined in Appendix A, will find useful application wherever minicomputers are used in NASA. Where options are of limited scope (e.g., business data processing enhancements) it is because of limits of the area of application, rather than of the scope of the tasks stated in the contract.

1.3 CHRONOLOGICAL SUMMARY OF ACTIVITIES

During October and November 1971, data processing activities were surveyed at Goddard Space Flight Center and the Manned Space Flight Center. NASA Headquarters personnel active in planning for or managing computer capabilities were also interviewed. Concurrently, a review of advances in information technology, current and forecast, was conducted.

By early December fourteen options were described to NASA for consideration. NASA reduced this number to the seven considered valid in the NASA context, and the following two months were spent researching these and documenting them more fully. A series of nine technical notes was produced during this period and ultimately published in Technical Reports 002 and 003 of this contract. The options are discussed in the following sections, below.

NASA selected three of the options as subjects for amplified reports outlining their development and application to NASA. Production of the three reports (Appendices A, B, and C) occupied the rest of the contract period.

1.4 RESULTS

The results of the contract are the three appended plans and the four options, which are described sufficiently for NASA to pursue. Together, they represent a development program for the next two to three years' activities to enhance NASA data processing capabilities.

1.5 ORGANIZATION OF THIS REPORT

The remaining sections of this report are devoted to a brief description of the seven options recommended to NASA:

Section 2, A NASA Family of Minicomputer Systems. This describes a development project to establish standards for minicomputer hardware and software in order to provide NASA with the economies of quantity purchases and interchangeability of minicomputer software, storage and peripherals. The standards define different minicomputer system components, each specialized for its intended NASA application, in as many levels of capacity as required. Section 2 is a condensation of Appendix A.

Section 3, Data Storage Technology - Hardware and Software. This describes the study of approaches to developing standard specifications for forthcoming, very large mass storage systems. The intent is to establish uniform standards for the hardware and software interfaces of the devices, eliminating much specialized programming and equipment. This requires determining the design parameters of storage systems best suited to NASA requirements. Section 3 is a condensation of Appendix B.

Section 4, A Program for the Systematic Evolution of a NASA Software Technology. This describes the development and application of a coherent software design and development technology under the guidance of a special technical leadership group at a pilot NASA computer center. A basic engineering approach employing techniques adapted from more mature industries as well as recently developed software techniques would be developed for new program production. The objective is fewer errors and more predictable costs of program development and more easily maintained and better documented programs (this section summarizes the contents of Appendix C).

Section 5, Data Processing Resource Reporting System. This recommends the establishment of a reporting system for any data processing resources used. It would facilitate accounting and budgeting for such necessary resources as computer time, programmer salaries, contractual efforts, etc. It would also support an inventory of equipment and of programs.

Section 6, NASA General Use Computer Network. Studies directed toward establishing a general use computer network exploiting the data communications capabilities available in the 1975 to 1985 decade are recommended. Standards for intercommunication languages and protocols would be developed based on NASA requirements for intercomputer communications and the plans of other government and industry groups for teleprocessing standards.

Section 7, Configurations for Processing Mission Control Data and Telemetered Experiment Data. A study is proposed to determine optimum configurations of computing resources for reducing and processing telemetered data in the 1975 to 1985 decade. A systems approach would be taken toward the various facilities now engaged in data processing in support of mission control and of spaceborne experiments. A desired result would be an optimum distribution of the three data processing entities: storage, communications, and processing facilities.

Section 8, An Integrated Business Data Processing System. This proposes the definition and design of a system to share business programs, data and computer resources among the various NASA centers and the Headquarters. Key centers would be designated to perform certain processes that are common to all centers. Electrical communications interconnecting the centers as necessary would provide access to the data. As a consequence certain data would become much more readily available (or available for the first time) at the Headquarters, and the cost of revising automated procedures would be greatly reduced.

SECTION 2. A NASA FAMILY OF MINICOMPUTER SYSTEMS

2.1 ORIGINS

The technological development that led to the minicomputer was the wide-spread commercial availability of integrated circuits in the mid-1960's. The advent of this electronic development made feasible the manufacture of computers at substantially lower cost and enabled independent manufacturers to jump quickly into the computer market with a significantly new product. In 1965, the minicomputer industry emerged with its first substantial sales of about $30 million. With rapid growth in sales, number of available machines and number of participating companies, the industry clearly established itself as a distinguishable segment of the computer industry. In the last half of the 1960's, annual sales have increased at an average rate of over 40 percent per year and, by the end of 1971, the installed base reached over 25,000 units.

The word "minicomputer" probably came from a paper presented at the Fall Joint Computer Conference in 1968, entitled "The Mini-Computer -- A New Approach to Computer Design." The term was applied to a host of

small general-purpose computers introduced during 1969 to satisfy the demands of the scientific, data communications, and control computer markets. By the end of 1969, a number of manufacturers also began to use "minicomputer" to classify small general-purpose computers aimed primarily at the commercial processing market.

AUERBACH Standard EDP Reports have roughly defined a minicomputer as a computer with the following characteristics:

- Costs less than $25,000 when introduced for a minimum stand-alone configuration that includes some type of input/output, such as a Teletype ASR 33 with paper tape attachment
- Provides at least 4K words of memory
- Performs calculations under stored-program control
- Can be programmed in an assembly or higher-level language
- Can be used by a broad range of users and is not restricted to specialized applications.

Although the preceding definition presents some difficulties, notably in minimum memory size and maximum price, and is arbitrary, it is sufficient to our present purposes.

2.2 THE TECHNOLOGY THAT MADE IT POSSIBLE

Minicomputer prices have been declining since before the term minicomputer was coined, or about 1963. This trend, which amounts to an annual decrease of 18%, occurred while performance as measured by the ratio of word length to main memory cycle time was increasing. The specific performance, or performance per dollar, displayed an uptrend of 50% a year. These trends reflect the revolution in semiconductor manufacturing technology which has occurred from 1965 to the present.

The upward trend in performance and downward spiral of minicomputer costs have been fueled by decreasing costs of logic circuits and

memory cores and by improved speeds of operation for the cores. For example, in 1965 a typical diode coupled transistor logic gate had a factory cost, when assembled on a printed circuit board, of about $2.70. In 1971 the same TTL gate packaged and mounted cost the factory $0.10, representing a compound reduction of 40% annually. During the same period the main memory core plane cost decreased from about 3.0¢ a bit to 0.5¢ per bit, equivalent to an annual drop of 27% a year.

Not only did memory core elements get cheaper during the era of the minicomputer, they also were improved to operate six times as fast. Whereas cores of 50 mil outside diameter, capable of memory cycle times of 6 to 8 microseconds, were in use prior to 1964, machines introduced since 1969 have utilized cores of approximately 20 mil o.d. and had cycle times of about a microsecond.

The effect of these changes is that minicomputers are going to get very cheap and hence very numerous. It has been estimated that minicomputers will account for about one-third of NASA's approximately 1200 computers by July 1973. While projections of the NASA inventory of minicomputers are not available, our best estimate is for 150 minicomputer acquisitions a year in 1973 increasing to 200 by 1975. Thus, NASA could have on the order of 900 minicomputers by the end of 1975 if acquired at a rate comparable to that predicted for the user community as a whole. What does this portend for large institutional users of computers such as NASA?

2.3 PROBLEMS IN THE USE OF MINICOMPUTERS

The need for careful specification of functional requirements is not limited to large scale data processing systems. The minicomputer shares this need, though on a smaller scale, with larger systems. It has been estimated that the costs of rectifying a blunder in specifying the ADP system to do a given job can exceed the cost of the equipment by a factor of ten. Even so, prospective minicomputer owners, faced with a

small budget for getting their new system up and operating, have in the past overlooked this crucial fact and slighted the important preliminary phases dealing with requirements definition and system specifications.

Many minicomputer applications have fallen outside the purview of the large-scale organizational computer center and its pool of personnel with computer-related skills. Minicomputers used in laboratories for scientific problem solving and data acquisition exemplify this class of applications. Users of these minicomputers may have scant previous experience with computers and be poorly equipped to develop programs on their new purchases. In the past, minicomputers - especially those at the low end of the size scale - typically afforded such novice programmers only an assembler, and lacked compilers and debugging packages to make his work easier. In addition, storage was limited and the loading of programs proceeded at the very slow rate of ten bytes per second of a paper tape reader. Since then, compilers have been added to the software offerings of most minicomputer manufacturers, and storage of relatively generous capacity is available, as are high speed input devices such as magnetic tape cassettes. These peripherals can run the cost of a minicomputer well up in the five figure range, however, and may defeat the objective of low-cost computing, especially if they are not needed for production runs of the programs, once written and debugged.

Hence the large-scale machine, if our minicomputer owner has access to one, offers an attractive alternative to writing programs on the minicomputer. Cross compilers, simulators, etc., operating on large-scale host computers for the purpose of producing code for minicomputers, have not been generally available for this purpose up to the time of writing.

The diversity of the minicomputer industry may present problems to the large institutional user. For example, the number of minicomputer manufacturers, which is declining somewhat now, hit a peak of over 60 in 1970; each produces several models. In addition, technological improvements cause the obsolescence and replacement of models in each manufacturer's line at intervals of two years or so. In 1970 Bell Telephone Laboratories

was reported to have 120 minicomputers in use, consisting of 34 models supplied by 12 different manufacturers; these numbers are undoubtedly larger by now. This raises questions of support: How can maintenance for this inhomogeneous brood be simplified? Can software development be shared? Can peripherals be swapped from one installation to another? How can one avoid building a special software interface for each mini requiring access to another computer? The larger the number of computing systems, the more pressing does the need for answers to these questions become, in order to use the investment efficiently.

Programming costs could also present problems to a large institutional user. Assuming for the moment a ratio of one programmer per owned minicomputer, NASA's estimated total of 900 minicomputers in 1975 would require annual programmers' salary costs in excess of $15 million. This figure, roughly 6% of the annual ADP budget, approximates the purchase price of the minicomputer hardware.

The particular problems arising from the use of minicomputers and singled out by NASA Headquarters for solution are those of high programming, design, maintenance and replacement costs due to the diversity of minicomputer models. In other words, is there something that can be done to contain the costs of programming minicomputers? And can the effort which goes into hand tailoring each mini installation somehow be minimized?

NASA has suggested a compatible family of minicomputers as a solution to these problems. Their compatibility would be such as to guarantee the interchangeability of machines of equivalent power, without the need to revise physical interfaces. Programs would also be able to operate on all minicomputer family members of power equivalent to or greater than that of the mini for which the program was written. The specification of such a family of minicomputers, their peripherals and software is the object of the development plan described below.

2.4 DESCRIPTION OF THE DEVELOPMENT

The basic problem to be solved in the subject development effort is that of establishing sufficient specifications and standards for minicomputer hardware and software to provide NASA with realizable economies in quantity purchases, interchangeability of minicomputers, software, storage and peripherals, and uniformly high quality. Inherent in this problem, as it is in the general problem of setting standards, is that of avoiding being more comprehensive and restrictive than necessary to achieve such goals.

The standardization achieved so far within the industry has been achieved basically for the benefit of the supplier, to provide an existing customer with a new generation product compatible with the old generation software and physical data files (tapes, cards). Because smaller manufacturers made their equipment compatible with that built by large firms, additional standardization was achieved which benefitted the user. In the proposed development project, the aim is to achieve still more standardization for the benefit of users, particularly NASA users.

2.4.1 Goals

The overall, long range goals of NASA with respect to the subject development effort are as follows:

- To achieve economy in purchasing minicomputers by "batching" its purchases according to some arbitrary size-speed categories and exploiting quantity discounts and other large-purchase advantages.
- To achieve flexibility, enhanced availability and economy by providing for reasonable interchangeability of both hardware and software, between manufacturers or suppliers, and between successive technological generations.
- To achieve measurable performance, quality and reliability of product so as to make these properties independent of supplier and technology.

- To achieve economy in operation and maintenance by providing modularity in design and commonality in spares provisioning and maintenance tools, instruments, procedures and training (at the system level).

- To achieve economy and speed in developing applications software by providing NASA users and their programming suppliers with large-computer power on large computers to develop completely ready-to-run small computer applications software.

The foregoing goals are qualitative, general statements of intent. In the next paragraph these general goals will be translated into objectives, which are rather more specific but are still long-range and fairly comprehensive. These objectives will then be examined more closely to identify the specific problems that must be solved in meeting each objective.

2.4.2 Objectives

- The hardware built to a given specification by various manufacturers will be interchangeable in a minicomputer system (e.g., a PA22 built by DEC will be interchangeable with one built by Data General).

- The functions and minimum performance requirements of a subsystem at a given level in the family will be incorporated in the specifications of the subsystem at the next higher level. Thus, equipment at a given level may be substituted for equipment at the next lower level of the family tree (e.g., a PA22 computer can replace a PA221, but not a PA213).

- Software designed to conform to a computer of given specification will run without modification on any computer built to that specification (e.g., an executive or application program will run on any manufacturer's PA121).

- Software designed to conform to a computer at a given specification will run without modification on a computer built to the next higher specification on the family tree.

- Program development software (assemblers, compilers, program aids, etc.) will be written for (designated) large (NASA) computer(s) to convert (designated) source languages to the machine language of any of the processors covered by the specifications, as designated at assembly/compile time

to the conversion program. The object is to provide a capability on a big machine to prepare ready-to-run programs for a minicomputer system, using all of the power and advantages that large machines have for normal (its own) program development software.

- Executive software will be written for each computer family tree, and will be comprehensive enough to cover all specified standard test system configurations. Executive software will be so partitioned that various subsets of modules can be used for appropriate levels of test system configuration complexity, for each level of the corresponding family tree. (It will be assumed that high-speed storage requirements will accompany each module, and that system response times can be estimated for various test system configurations using standard tests.)

- Standard tests will be designed and standard test data prepared, with the correct values being provided and tested by the test software. Bench mark programs will, therefore, not only provide general system check-out and running or response times, but also accuracy and precision (if appropriate) of test results. Instruments to test electrical and electronic circuit responses to test programs will be included. Test programs will be designed for various standard test configuration at all size-speed levels.

- Standard procedures and tests for measurement of reliability, maintainability and repairability will be developed.

2.4.3 Activities Required for Development of Product-Family Guidelines and Standards

The proposed development work is not unlike a commercial product-line development program in its objectives. The principal difference is that in a product-line development, the supplier is interested in interchangeability and compatibility in order to provide his customer with as complete a line as possible, and thus to capture the largest possible share of the customer's current and future business. In the proposed development project, the sponsor - NASA - is interested in interchangeability and compatibility between the product-families of different suppliers of the same kind of product, as well as between categories of products and successive generations.

There are several complementary activities of primary importance in a commercial product-line development program. These include, in approximate order of importance:

- Market Research
- Technical Research
- Intelligence - (Competitor Activity)
- Field Service.

There will be activities analogous to these in the development project. In the place of Market Research, for example, the activity required will be to examine current and future applications of small scale computers, within NASA, to develop from this information the types of products needed and to make estimates of the quantity of each kind that will be required.

Technical research covers several subordinate activities: circuit technology, systems technology, manufacturing technology, and programming technology. The role of these technical activities will be very much the same as in commercial product development. The activity analogous to intelligence on the competition will be intelligence on the present and planned product lines of industry, in both hardware and software. Corresponding to Field Service, NASA will have to consider providing maintenance of its user groups, including spares provisioning, maintenance personnel, and diagnostic, test and repair equipment.

2.5 THE MINICOMPUTER FAMILY - AN EXAMPLE

The intent of the structure we present here is to serve as an example of what might be produced in a fairly large-scale, intensive standards development program, and not as a first iteration of what will be produced in such an effort. At the same time, we must admit to the fact that the structure presented contains some "editorial" content - some opinion. That opinion is expressed in a number of ways; viz.:

- The structure is functional. It is not based on technology, embodiment, or size and speed
- The structure is based on the use of standard modules and a unified bus; that is, a bus to which subsystems may be logically connected, and along which travel signals of standard size and format.

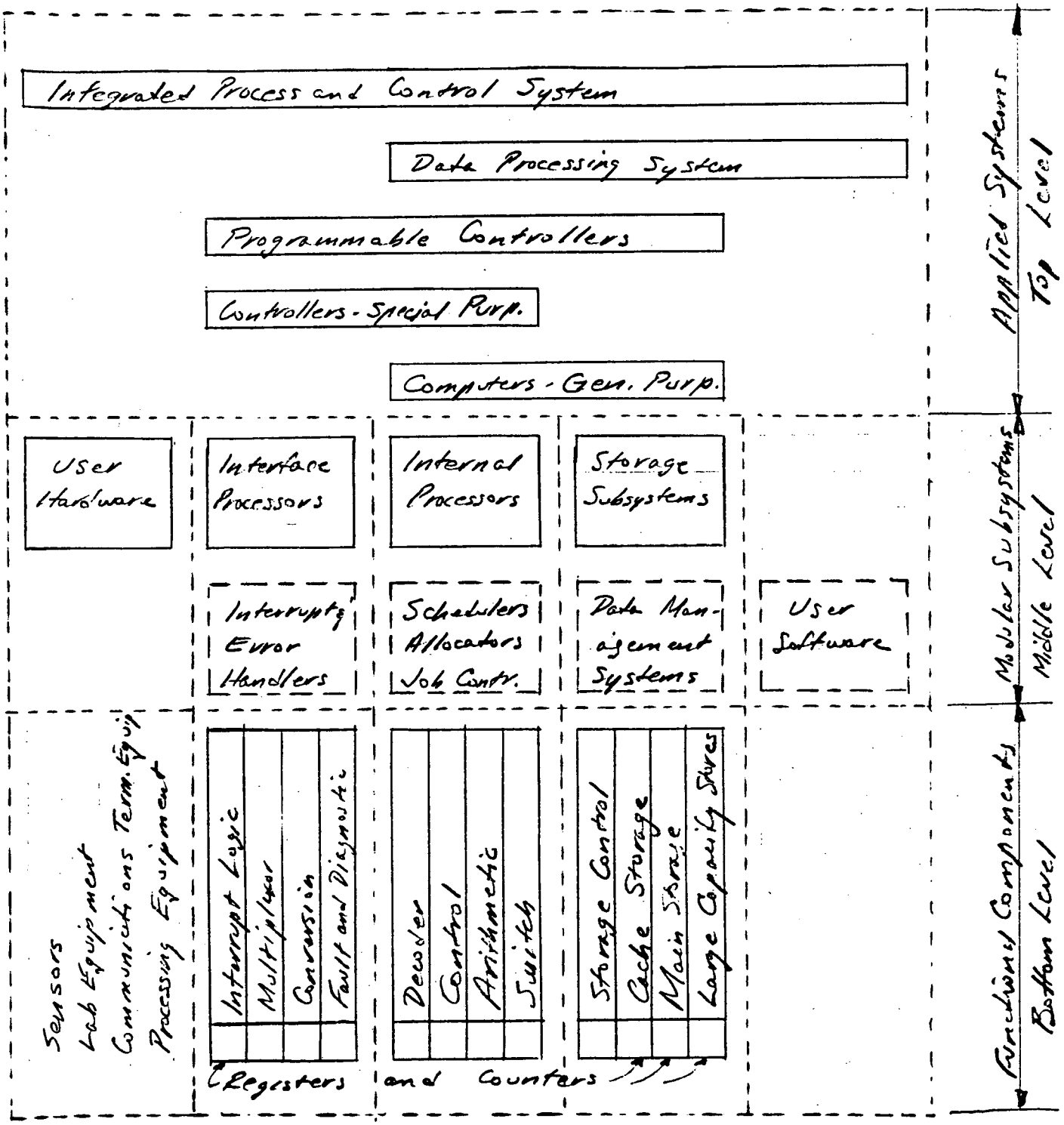
However, it must be emphasized that the example is not provided as a vehicle for the expression of our opinion; the main purpose of the opinion is to provide reasonable substance to the example. A lesser purpose is to provide a reasonable take-off point, or "straw-man", for a full-fledged development effort.

2.5.1 The Family Tree

The structure is presented in the accompanying diagram. It shows three functional levels: the applied systems level, the modular subsystem level, and the functional component level. The top level, applied systems, includes general purpose minicomputers which are made up of internal processors and storage subsystems only. It also includes special purpose controllers which would cover most current peripheral controllers, and programmable controllers, which include read/write memory. The structure thus provides the possibility of including one or more standard minicomputers (of a range of sizes) and programmable controller configurations, as well as micro-programmed special-purpose equipment with read-only memories.

At the middle, or modular subsystem level, most standard items of hardware and software would be established. These modular building blocks, designed to interface on one or more unified busses, together with standard system software packages, will be used to build up systems of the desired nature, complexity, throughput and response time. Although upper ranges of size and speed are not precluded, we envision large throughputs and short response times in applied systems being achieved through parallelism rather than through inclusion of very high-speed or large capacity family members.

The basic elements of this middle level include the internal processors, storage subsystems, interface processors, and the user's hardware and software. We are concerned here primarily with the first three of these.



Structure of family of computers, subsystems, and functional components.

By internal processors we mean the hardware or firmware (read-only memories and micro-programs) more commonly termed Central Processing Unit. By interface processors we mean units driving user equipment external to the computer system. The types of processing that might be included in this area are analog-to-digital conversion (ADC) from sensors; digital-to-analog conversion (DAC) for control of process equipment; transmission line protocol, error detection and correction, and correct transmission acknowledgement-no-acknowledgement (ACK-NAK) of communications equipment; data acquisition and communication line multiplexing or concentrating.

Storage subsystems, again categorized functionally rather than physically or technologically, include stacks, cache memories, main stores, large-capacity stores, on-line stores, and archival stores in which permanently recorded data can be taken off or put on line (tapes, discpacks, possibly photographic plates).

In each of the foregoing instances, the modular aspects are stressed, so that modules can be added or removed to change the size or capacity of a system, and new technology modules of a given function can replace obsolete examples of the same function without providing special interface hardware or software.

The bottom level of the family comprises assemblies that go to make up the modular subsystems. The technical or physical embodiments of these components are not relevant. It is the intent to make the structure as independent of technology as possible, rather than to base the structure on a particular technology. Furthermore, by using as the foundation of the structure a standard bus, or set of busses and switches, and an indefinitely large set of modules, we have made the structure independent of an organization or control hierarchy.

SECTION 3. DATA STORAGE TECHNOLOGY - HARDWARE AND SOFTWARE

3.1 INTRODUCTION

The optimal design of a mass storage system, its efficient utilization during computation, the providing of useful data services to the programmer, and the effective handling of data logistics during program operations are some of the most important, and least satisfactorily answered, questions facing large-scale computer users at this time. These are the issues of data management, using the phrase in its broadest sense, and because they are related they deserve comprehensive analysis as a set of related issues. A concerted attack on these problems may yield the most powerful solutions and such an attack, in fact, is planned in the study proposed in this Section. For convenience, the study is divided into two facets, storage management, and data management. The problems of the design and selection of a storage system and the handling of physical space allocation and physical data storage logisitcs are treated within the first task (storage management), while the problems of the programmer interface and logical data structures are treated within the second task (data management).

The storage management function comprises the specification and selection of a set of storage devices with adequate capacity for the installation and a strategy (and possibly a hardware storage processor) for allocating and managing storage space. The storage management function becomes critical when the total capacity required exceeds the capacity of fast-access disk systems and the responsiveness required can neither be compromised nor can it be satisfied by traditional manual re-handling approaches. There are several instances in NASA where fast (real-time) response is required in installations which manage data volumes in the trillion bit range. To meet these demands requires utilization of the latest terabit (10^2 bit) capacity mass storage devices and advanced level-changing strategies in multi-level (hierarchic) storage systems. By level-changing is meant that transfer of data from slow access to faster access storage. A subsidiary need is to take full advantage of recent advances in storage devices and storage management algorithms, and perhaps to specify needs not satisfied at the moment in order to channel further research in storage devices and strategies into the most productive areas.

Research in information storage has spawned two new technology areas which promise devices with characteristics distinctly different from those available at the present time. The first area, which can be called bit transfer devices, promises to fill the space-time performance gap between magnetic core and drum/disk memories. The second area is optical storage of information in the form of a hologram or intereference pattern. This second area presents the unique capability of providing parallel access to a page of data in microseconds. The emergence of new storage device technologies such as bit transfer devices (which include magnetic domain devices, charge coupled devices, and others) and holographic stores significantly alters contraints on access time and capacity so that storage hierarchies of much more flexible characteristics can become available.

As a result of this situation, the use of a hierarchy of storage devices utilizing several technologies will remain the only effective solution to providing optimal performance in those computer installations which must provide access to an extremely large volume of data. In order to realize the promise of high performance in a hierarchic storage configuration a complex

logistical problem of storage allocation and data movement across levels of storage must be solved. This problem has indeed been solved in specific installations and it is conjectured that a general solution, insensitive to specific computer and storage device types, is possible. When realized, this hardware/software processor, called the Storage Management System, will automatically allocate storage to data and change the highest storage level allocated to a specific data entity (page, or segment) either as a result of a specific program reference to a data entity, or in anticipation of such reference. This process of dynamic "level changing" will be a prime function of the Storage Management System.

Several considerations argue for establishing a single storage management system as standard for NASA. First, are the savings which can be made over developing management systems for each NASA mass storage installation. Second, is the ready accessibility to data afforded remote users by a standard language and standard data management services, data types, etc. Third is the increased effectiveness of program development activities due to the use of standard, powerful data management services. And finally, providing standardized environments for program execution goes a long way toward achieving program and data transferability across installations and machine types.

In order to design a data/storage management system, the performance and control characteristics of the storage devices must be known, and a host processor must exist which can provide the necessary control signals. We assume the latter condition will be met by processors, modified if necessary, available at the time the devices make their production appearance. The former - knowledge of the devices' characteristics - is not subject to such an assumption owing to the present developmental nature of the applicable technology. The project takes a different approach, that of specifying the characteristics it would be desirable for the devices to have. The approach is not an unreasonable one for the bounds for the characteristics can be deduced from physical principles.

3.2 PURPOSE OF THE DEVELOPMENT PROJECT

The purpose of this project is to specify the components of a storage management system employing devices of advanced performance and massive capacity. The intent is to permit the development of a NASA standard system, which avoids the cost of developing individual mass data storage systems for each installation and eliminates differences in the procedures for automatic access to them. The system components, and the related development to be undertaken by the project are:

- Storage Devices. A hierarchy of storage equipment, ranging across the speed/capacity/cost spectrum, and consisting of items of standard manufacture will be used. Performance specifications for the devices required will be developed by this project.

- Storage Management Processes. This comprises the processess necessary to the logistics of moving data automatically between slower- and faster-access storage, which may be executed by hardware dedicated to the purpose or shared by other processes. The project will develop effective strategies, and a system architecture, for carrying out these storage management functions.

- Data Management Processes. These consist of programs to provide data management services, such as establishing a file, reading into it, writing from it, etc. The project will specify a set of data types suitable for general NASA use and a Data Management System architecture which supplies a full range of necessary services for these types.

- User Language. The project will specify a language for invoking the services of the Data Management System.

The project will also develop a storage system simulator, which will be used to determine the performance of a given mix of devices in a storage hierarchy and/or a given logistics strategy employed by the storage manager.

3.3 PRINCIPLE BENEFITS AND BENEFICIARIES OF THE RESULTS

The obvious areas to benefit from the use of a systematically specified set of storage devices are NASA's ground-based computing facilities, particularly those requiring access to large, on line data bases. These results of a uniform set of performance specifications are likely:

- A set of storage devices which more closely meets the needs of NASA is apt to become available.

- The Data Management System's paging function, which provides for automatic data movement across storage levels, will be more effectively designed, implemented and debugged.

- The way the Data Management System interfaces with the storage device controller will be designed once and will not have to be modified for each device or installation.

The use of the same data management system at several mass storage installations means that the stored data bases will be accessible in identical ways. Thus computer programs used to interact with data stored at any one of the installations will work equally well with all, aside from substantive differences in the data itself. This greatly simplifies program development for a single consumer of data which is stored at several of the sites, all using the standard Data Management System.

The availability of a standardized interface which provides data services at a number of levels, appropriate to a number of user types can have a profound effect on the cost of new program development for NASA. It is almost trite to say that the problem of data logistics is one of the most dominant problems in computing today. Every program, whether application program, compiler, or query interpreter, requires these services, hence every programmer must solve a data logistics problem. If these services are provided in a centralized, standardized, way in the computing facility's operating system then a large proportion of the cost of program design, implementation, debugging, and maintenance can be avoided.

SECTION 4. A PROGRAM FOR THE SYSTEMATIC EVOLUTION OF A NASA SOFTWARE TECHNOLOGY

4.1 INTRODUCTION

Computer programs from the very first have been subject to errors - missteps in coding, perpetrated by the programmer and not found until after the results of the program's operation are examined and seen to be in error. Errors may be obvious or elusive, but in either case they have to be diagnosed after the fact, for the computer proceeds at such a pace as to make concurrent diagnosis out of the question. The human tendency of programmers to err is with us in undiminished form today as it was at the inception of the stored program electronic computer twenty-five years ago.

Programmers seem to be unable to estimate the size or the difficulty of writing a program which they have never attempted before. This becomes highly undesirable in large programming projects, requiring dozens or even hundreds of programmers, which therefore have a tendency to miss their scheduled target dates and costs by wide margins. Unfortunately the miss is usually in the direction of an overrun, a fact attributed to

inefficiencies due to the large organizations required and a source of discomfiture to project managers.

SAGE is regarded as the first large scale complex programming system; a thousand people were involved in its development. Based on a prototype system developed at MIT, the full size system should have required a reasonable number of people and time, but more effort - orders of magnitude more in fact - were needed. Various specialists were required at all levels of the program. All of these specialists required managers, themselves at a variety of levels. The managers required help, both administrative and technical in nature. As schedules tended to slip or difficulties be recognized, more people were hired which required more management (and more communication). This cycle continued for several years until many hundreds of people were involved in the programming effort. The program, considered by most people to be a landmark as well as one of the few successes in large scale system programming, nevertheless was delivered later than originally planned and with somewhat less capability. When asked what he would do differently if he had to do a system like SAGE again, the manager of SAGE development said, after some reflection, that he would hire twelve good people to do the whole job. Outside of that he couldn't think of much else that he would have done differently.

In the words of one observer:*

> These problems are symptomatic of the lack of an adequate basis in the methodology, technology, and theory of information systems and/or a lack of disciplined application of the methodology and technology we do possess. We are cursed with the problem of the large, complex system - problems of dimensionality and scale - for which there is neither an adequate science nor an adequate engineering discipline.
>
> Too often trial and error is the practiced methodology to match an information processing system to the need. The heuristic approach is still the rule rather than the exception in a computer systems design.
>
> In defining the information system requirements, frequently the real problem is not clearly known or, even worse, is incorrectly defined. As a result good solutions are formulated to

*I. Auerbach, in address to 10th Anniversary meeting of IFIP, Amsterdam, Netherlands.

wrong problems. System specifications may propagate incorrect problem definitions that are biased by the designer's experience so that they will reflect the limitations and errors of other systems. Empirical solutions are frequently "force-fits," and inefficient solutions to the problem.

In this section a program is described whose ultimate purpose is to make possible the production of software in NASA within predictable schedule and budget constraints and with major characteristics - such as size, run-time, and correctness - predictable within reasonable tolerances. As part of the program a pilot NASA computer center will be chosen to apply software development and management techniques systematically and determine a set which is effective. The techniques will be developed by a Technology Group, which will guide the pilot project and be responsible for its success. The application of the technology will involve a sequence of NASA programming tasks graduated from simpler ones at first to complex systems in late phases of the project. The evaluation of the technology will be made by monitoring the operation of the software at the users' installations. In this way a coherent discipline for software design, production, maintenance and management will be evolved.

4.2 PROBLEMS WITH SOFTWARE DEVELOPMENT

4.2.1 Lack of System Description Languages

Apart from informal languages, which arise more or less on an ad hoc basis, there is no reasonably concise and unambiguous language in general use to convey the meaning of computation processes among humans. Natural language, flow charts, and higher level programming languages are frequently used, but their use involves the possibility of misunderstanding.

Natural language is the medium generally used for communication with the user about his requirements. In such language, the danger of misunderstanding is great. Numerous examples of information systems which failed to answer the needs for which they were designed can be cited as proof of the need for a less ambiguous communications medium.

Language also has an effect on the thought processes of those who use it, and the particular design language employed by a computer systems engineer will influence the character of the design he produces. It has been shown that a design team must first agree on a common language suited to its project before it can progress with the design.

4.2.2 Test of Correctness Impossible

A computing process can be viewed as a succession of machine states dictated by the input data. It has been shown that the number of possible input sequences, and hence the number of possible states, is so great that it would take tens or even hundreds of human life spans to demonstrate them all on a computer of practicable speed. While it is possible to test the logic flow of a program in finite time, demonstrating the correspondence of the output to that required is what would take impossibly long. This obstacle constitutes a gulf separating the design of computation processes from that of physical entities: no formal check can be made of the correctness of a design. Designers must use informal methods, at least until algorithms for formal proof are perfected. Currently, human intelligence is the only means available to check the correctness of programs. Programs must be concisely expressed to remain within the limits of human understanding.

4.2.3 Programmer Psychology

Programming and more especially systems design, is acknowledged to be a creative activity and attracts creative people. However, the lesser aspects of programming (the production of "dull" sections of code) do not appeal to the creative sense. In consequence, design functions tend to be distributed among all the programmers on the project as compensation for purely production coding, with resulting lack of control over the design process. The lack of restraints on designers' inventiveness, such as a deadline or firm system goals, has been known to cause programming project failure.

Programmers identify with the code they produce, to the extent that errors in code tend to be glossed over by its inventor. Once the programmer's ego is divorced from his code, the errors become highly visible, and in fact "egoless programming" is a term which describes the practice of critical review of code by the programmer's peers.

4.2.4 Precipitate Coding

The pressure of a schedule and awareness that a great deal of coding has to be done has caused managers to commence work on coding just to get started on a job which is obviously huge. When combined with an organizational philosophy which puts coders at the bottom of the management structure, this hasty commencement of coding throughout the system leads to design difficulties. We recall that even at the coder level some design latitude is allowed as a compensation for the dullness of mere coding. Hence the process of design is commenced throughout the system at the very bottom level by the coders before the design has been properly thought out. A classical bottom-up design emerges, leading to difficulty in integrating the resulting components into a system, but its most serious drawback is that the resulting system design itself may be influenced by the existence of modules already coded.

4.2.5 Programmer Training and Selection

Software design principles are largely untaught in courses for programmers, or elsewhere. The burden of what is taught is how to use a programming language, with the implication that design ability is conferred with mastery of the language and consists simply of employing it correctly. It is generally acknowledged that programmer aptitude tests distinguish not between poor and good prospective programmers, but more nearly how these programmers will do in training or how easily they will learn programming. Although college degrees have been required for programmer recruits, no correlation has been shown between the quality of programs produced and the amount of such education received, except in scientific programming which requires a knowledge of advanced mathematics. It has been acknowledged that the identifying characteristics of potentially good programmers have not yet been isolated.

4.3 THE APPROACH TO THE SOLUTION

The existence of problems has been recognized outside the United States. In 1968 and 1969, conferences were convened in Europe by the NATO Science Committee to define the problems better and try to find solutions, for which a special term, software engineering, was coined. Those attending the international software engineering conferences raised many problems, aired many opinions, and presented many excellent ideas for solutions to some of the problems. However, the primary goal for virtually all of these attendees is that of producing software to perform the functions intended by them or their clients. The problems attendant on such production and their solution are therefore of secondary importance to them. This implies a lack of comprehensive, systematic and sustained efforts to solve the overall problems that beset the software-producing industry; and upon closer examination of the industry, such effort is indeed missing.

The proposed program is specifically advanced to fill this gap; its sole objective is to provide graphic and verbal languages, procedures, constructs, models, organizations, documentation, specifications, job descriptions, test plans and various kinds of standards. That such an effort is needed is attested to by both the title and existence of the Software Engineering Symposia. They comprise a recognition by the leaders in the industry that an engineering-like approach is essential to the vigorous growth of the software industry. At the same time, the difficulty of achieving such an approach, because of the nature of the end-product, is also recognized. Thus, the proposed program.

The present proposed effort, in effect, picks up where these conferences left off: the formation of a group dedicated to solve the problems of software production over an extended period of time. The planning, launching and execution of such an effort requires both the resources and the promise of large payoff that apply only to an organization of the size and scope of NASA. The autonomous nature of the organizational entities of NASA will assure that the success of such an effort will depend on its merits rather than on the authority of its sponsor, NASA Headquarters.

4.4 GOALS AND OBJECTIVES

A program of the sort proposed is appropriate to an organization of the size and mission of NASA. It takes a large organization to have a sufficiently great stake in such general and long-range goals as those proposed for this program. A small organization simply cannot afford to take a global or long-range view; satisfying immediate needs is all it can afford, and generally this is sufficient. Such a situation applies to most of the parent organizations of the attendees of the Software Engineering symposia.

The mission of NASA involves a truly incredible array of computing power, from the smallest computer to the very largest complex, from the slowest processor and memory to the fastest, and from the most accessible to the most remote. It also involves an unprecedented array of applications. Altogether, there is little in hardware, software, or application that is not represented in a significant way in NASA centers or by NASA users.

The organization of NASA is uniquely appropriate in that the source and mechanism for the special funding and subsidies that may be required exist, and at the same time the autonomy of individual centers and users is such that the program can proceed with a maximum of freedom and virtually no bias or explicit technical direction from the top. There is enough variation in software development practice that objective criticism can be expected. The fact that it is a government organization, at the same time a user of enormous size and influence, and one with clearly no vested interest in specific hardware or software producers is also significant.

The goals and objectives of such a large and diverse organization will be sufficiently global and general that no effort need be made to keep it from having a uniquely NASA flavor. Conversely, there will be no difficulty in interpreting goals and objectives stated in general terms to specific NASA or center interests. Thus, the goals and objectives that follow are general.

4.4.1 Goals

The long range goals of the proposed program, that is, of the Software Technology it is designed to establish, are as follows:

1. To fulfill the user's requirements and expectations in the end product with respect to usability, usefulness, cost and time;

2. To produce an end product satisfying goal 1 and having predictable characteristics such as modularity, size, run-time, response time, numerical resolution, and correctness;

3. To produce an end product which makes measurably efficient use of available resources both in the process of its production and in its structure and operation; and

4. To establish quantifiable parameters for describing the properties of computer software and firmware, develop means for measuring the value of these parameters in specific instances, and develop procedures for applying these techniques in assuring goals 1, 2, and 3.

These qualitative and general statements of intent can be broken down into more detailed objectives. These are defined in the next section.

4.4.2 Objectives

The objectives stated below start with the perspective of an entire computer system, and then consider hardware and software individually. Actually, the program in the very long run includes a gradual expansion of scope to include firmware and hardware. It could, of course, include data transmission and communications at some point and to some extent and depth best determined by those involved in the program.

Additional objectives could be defined. More detailed objectives touching on explicit design, fabrication, and test procedures could, for example, be added. Those listed below will be sufficient for the present purpose.

1. The user will be able to describe his functional, procedural and data problems to a computer systems engineer who will express them explicitly and rigorously in documentation comprehensible to the user or his agent.

2. The computer systems engineer will be able to translate the functional, procedural and data aspects of the user's problem into structural terms using standard verbal and graphical languages and appropriate measurements.

3. An arbitrarily selected computer systems engineer of established reputation and competence will be able to review the planned structure of the proposed computer system, hardware, software and firmware, and certify its structural integrity; and examine the functional, procedural, and data descriptions, and certify that the planned structure and data sources will be adequate to accomplish the required functions and procedures. (Verify preliminary design.)

4. Computer Design Engineers of various specialties (hardware, software, firmware) at successively lower levels will be able to generate designs and/or specifications to correspondingly lower levels of detail, using standard "parts" wherever possible.

5. Computer Engineers and Technicians of various categories and levels will be able to schedule, fabricate and test individual modules, and assemble and test them in successively higher levels of structural and functional assemblies. (This applies separately and collectively to hardware, software, and firmware.)

6. It will be possible to include in the designs and specifications at all levels any values of various numerical parameters: for each component part, the manhours and elapsed time to design, fabricate and test; and for each testable component, performance measurements that can be traced back through the structural hierarchy to the user's requirement: execute time, response time, propagate time, throughput for various defined initial load conditions.

7. The product at any stage of completion (including designs and specifications) can be measured and meaningfully compared quantitatively with the requirements and design parameters of higher levels.

8. It will be possible to establish procedures for checking and approving components and assemblies at all stages of design and fabrication; the object will be to permit establishing responsibility and accountability for deficiencies or errors.

9. The establishment of positions of defined responsibility and defined procedures and standards will make it possible to establish well-structured general and special organizations capable of exerting effective management control upon projects and upon their funding and scheduling.

The way in which the program is organized to achieve these objectives will be discussed in the succeeding sections.

4.5 THE ORGANIZATION AND OPERATION OF THE PROPOSED PROGRAM

The basic requirements for the program are (1) that it provide not only for developing the technology, but for applying, testing, and evaluating the results as well; (2) that the responsibilities for development, application and evaluation be assigned to separate groups; and (3) that the development be evolutionary, that is, that the technology be applied to successively larger and more complex problems, and modified and improved after each application. There is further the longer-range desirability of merging the software technology with that of computer hardware, computer systems, communications systems, and information systems.

The responsibilities for development and application of the technology and of evaluation of the results will be assigned to three separate groups which we shall call, respectively, the Technology Group, the Software Group, and the User Group. These will be discussed in more detail in the next section.

Each of the succession of applications of the technology will be called a cycle. A new software development problem will be undertaken in each cycle, the nature of which will be determined at the conclusion of the preceding cycle. Each cycle will be divided into three major phases: Phase A, in which the Technology Group will amend or modify the technology as a result of the previous cycle, and select a specific software development project from among those coming up to test the new version of the technology; Phase B, in which the Software Group applies

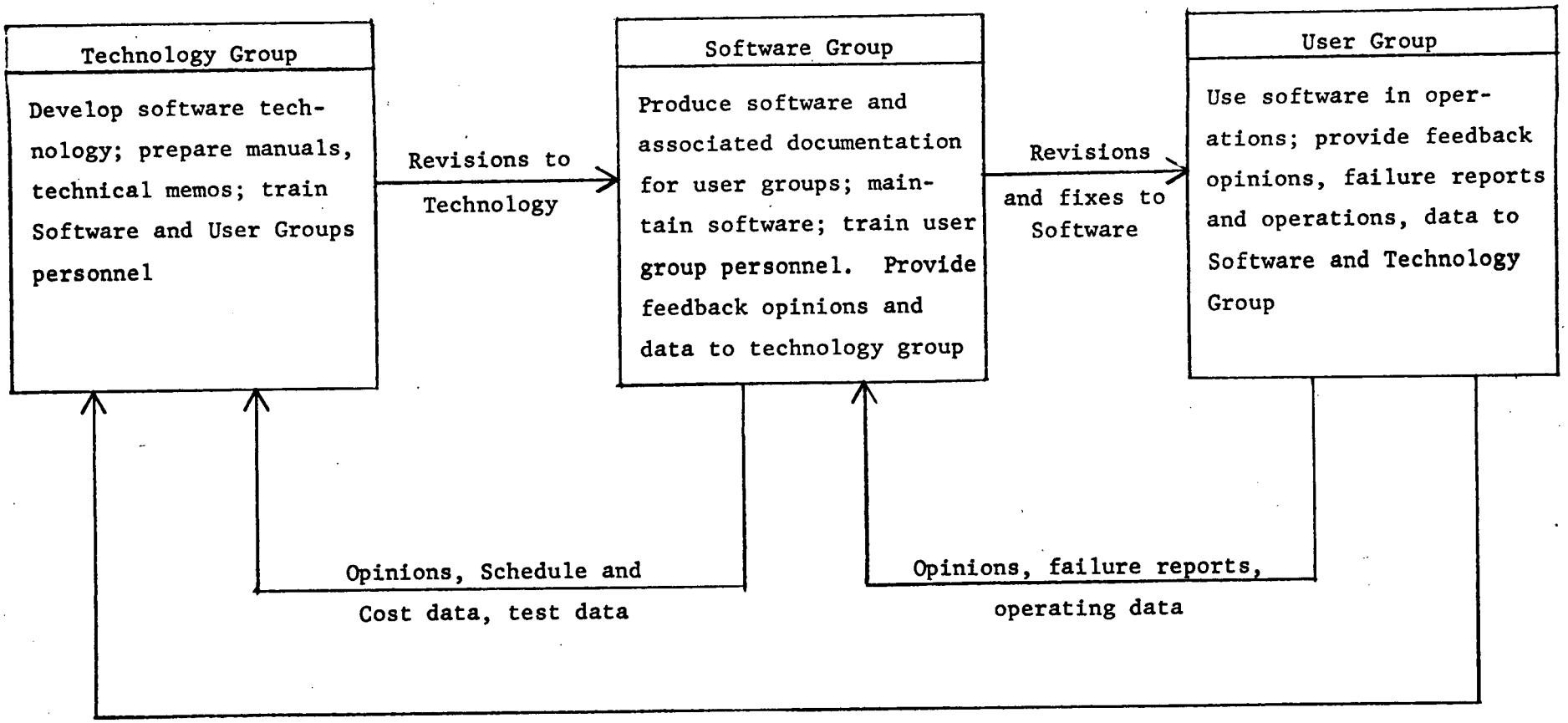
the new version of the technology and develops the software package selected by the Technology Group; and Phase C, in which the User Group will operate with the new software which will have been developed for them. These phases are discussed in more detail in the next section.

4.5.1 The Participating Groups

Although the three participating groups will be separate and will probably even lie within separate higher-level organizations, it is essential that they cooperate fully and coordinate on pretty much a continuous basis. This is shown in Figure 4-1. Aside from the fact that both relative independence and a community of interest between the three groups can exist at the same time (because of the overall NASA sponsorship), little can be said at this time about formal organization. It is possible, however, to say something about the general nature of each group.

The Technology Group. We suggest that this group be sponsored and funded directly by NASA Headquarters. In this way, the long range plans and policies of NASA will be able to influence directly the development of a software technology responsive to NASA needs. The initial group should be limited to four or five very senior people. In later stages of the program, the group might be augmented with people drawn from the Software Group and the User Groups. Such additions would bring directly to bear the personal experiences of those on the receiving end of the new software technology. Also, in later stages, as the group turns increasing attention to microprogramming and hardware, people experienced in the application of these areas should be considered.

The Software Group. The Software Group will be selected from one of possibly several operating within the NASA center which will have indicated its interest in participating in the program. The Technology Group, formed first, will make it one of its first tasks to determine the desirable characteristics of such a group. Center personnel can then be interviewed for their interest and opinions. There is little doubt that



Technology Group: Sponsored & funded by NASA Headquarters

Software Group: Sponsored & funded by NASA Center

User Group: Any user group

Figure 4-1. Participants in the Software Technology Program.

the project will attract unusual attention, and that there will be great interest on the part of existing software organizations to participate in the project and play a creative role. The Software Group will be sponsored and funded in the normal way by its Center; however, extra funding from NASA headquarters to support certain experimental or risk aspects of the program might be in order. It is possible that initially a separate Experimental Software Engineering section should be established to work on the program, rather than try to reorganize an entire software group. As the emerging technology proves itself, the size of the experimental group can be increased at the expense of the established group until the desirability of complete cutover to the new approach is apparent.

The User Groups. If the initial Software Group is an integral part of an existing software shop within a Center, user groups will comprise the normal clientele of that Center and that shop. The desirable characteristics of the participating Center might well include the nature of its mission, clientele, problems, and organization. Thus, the user groups would use their normal funds to secure their normal software services. Consideration for some extra funding, made available from NASA Headquarters through the Center for extra services or manpower for the Software Group, might be given to expand or modify software problems slightly to make them more appropriate vehicles for program objectives. It will be desirable that the mechanisms for such supporting funding be already established.

There is, of course, no reason that the experimental group should not undertake more than one problem at a time, provided that one and only one problem at a time be undertaken of an untried size and complexity. It would be in order for the Software Group to undertake programs of a size and nature that it has demonstrated it - and the new technology - can handle. However, the same careful follow-up and feedback through the software design, fabrication, test, operation and maintenance stages should be observed by all groups. It is to support those activities that extra funding might be arranged for the software and cooperating user groups.

4.5.2 The Phases and the Actions and Interactions of the Three Groups

Although the projects for the cycles will be different, the phases within each cycle, and the activities of each group in each such phase, will remain very much the same. The phases are as follows:

- Phase A. Technology Research and Development
- Phase B. Software Design and Fabrication
- Phase C. Software Operation and Maintenance

Phase A. Research and Development (R&D Phase). In this phase, the activities of the Technology Group will dominate. This group will examine the chronic problems of software development and will conduct research on software development methods, techniques, approaches, organizations, etc. that have been advanced to solve the problems. It will then develop an overall approach to the solution of these problems, comprising descriptions of an organization for designing, fabricating and testing software (and firmware), descriptions of the staff positions in the organization, procedures, techniques languages and graphics, standards and estimating methods. It will then assist in the organization and staffing of the Software Group, and cooperate with this group in selecting a user and his problem as its first or next effort. It will also establish tentative communications with the User Group so that it can obtain independent and objective information from both Software and User Groups on their respective problems and experience.

The Software Group will refine its internal organization during this phase and make preparations for using the revisions to the technology being developed by the Technology Group. It will interact with the Technology Group all during this phase by giving its reactions and opinions to the additions and modifications to the technology being planned by the Technology Group. It will also cooperate closely with the Technology Group in selecting a specific software development task from among those being presented to it by its users.

When an application has been selected, the sponsoring User Group will establish working relations with the Technology Group, and become familiar with the kind of information desired as feedback. Examples are the effectiveness of its communications with the Software Group, the latter's responsiveness to and comprehension of its needs, its ability to read and interpret preliminary designs and specifications, its reaction to various functional performance and acceptance tests, and finally, its reaction to the effectiveness of the software product itself and associated documentation, training, maintenance, and so on.

Phase B. Software Design and Fabrication (D&F Phase). The Computer Group dominates in this phase, in which it will work with the User Group in explicitly and unambiguously defining the problem, establishing user constraints (time, cost, environment, operating and using personnel and specifications), developing a preliminary design for approval by the user, and the subsequent detail design, fabrication and test of the system.

The User Group will work with the Software Group to develop the requirements and preliminary design, and again in monitoring performance and acceptance tests on the major assemblies and completed system.

The Technology Group will observe and coordinate with the Software Group for deficiencies or weaknesses in the organizational structure, job functions, languages, etc. as the development work proceeds. It will not in general be concerned with assessing the quality of the product; rather, it will be concerned with such matters as lack of communication or understanding, schedule delays and slippages, missed estimates on man-hours, interface or system integration (assembly of parts) problems and the like. It will also coordinate with the User Group to obtain its reactions and opinions on the responsiveness of the Systems Group to its needs and its opinion of tests on major assemblies and subsystems.

Phase C. Software Operation and Maintenance Phase (The O&M Phase). In this phase, the activities of the User Group will dominate. The software developed by the Software Group will have been delivered in this phase, and its operation and maintenance will have begun.

It is assumed that the user will make his own arrangements for operation and maintenance, and that adequate documentation for this purpose will be prepared by the Software Group, accompanied by training of operating and maintenance personnel. However, it is also assumed that the Software Group, as part of its contract, will be responsible for some post-installation warranty service.

There will therefore be some communications between the User and Software Group during this phase. As a matter of fact, there probably ought to be arrangements for failure reporting for an extended period after installation--enough time for failures to settle down to a "steady state." There will also be communication, for about the same "extended period" mentioned above, between the User and the Technology Groups. Such continuous cooperation and communication during this phase is most important. The Software Group will be interested in user feedback to modify its design to minimize warranty costs and customer maintenance; the Technology Group will be interested in user feedback to see how well the user anticipated his own needs and wants, how well he expressed them to the Software Group, how well the design engineers translated these to structure, and how well the fabricators within the Software Group were able to follow the design, how usable the software was by the user's operating personnel, and how useful the system was for the user's top management. This feedback will be used by the Technology Group to recommend changes to the organization, procedures, etc. of the Software Group for another cycle and another problem.

4.6 THE ORGANIZATION AND OPERATION OF THE TECHNOLOGY GROUP

Given the general goals and objectives relative to the technology to be developed as stated in Paragraph 4.4, and given the purposes and missions relative to the NASA sponsors of the program, some reasonable conclusions can be drawn about the organization and operation of the Technology Group. These conclusions will be described and discussed in this section, with expansion of several relatively important topics in later sections.

4.6.1 Size and Composition

The group initially should be quite small - perhaps four or five persons. The formal organizational structure at this stage will therefore be inconsequential - a leader, three or four innovative, top-level people having fairly broad systems backgrounds with a concentration in computers and non-trivial programming experience, and one or two support personnel.

4.6.2 Initial Role of the Technology Group

The role of this initial group will be that of advisors and consultants to both the Software and User Groups. In fact, the problems of these two groups, as perceived by them, in developing and using software (respectively), comprise perhaps a better set of initial tasks than an ideally-defined set derived from a study of the industry at large. Careful analysis of such problems, developed through informal interview, interaction and discussion, will reveal areas within the goals and objectives of the program. Quick solution or assistance in small, irritating but perhaps superficial problem areas will establish credibility, trust and a good rapport much sooner and easier than deeper and more subtle problems.

The initial role, then, will not be that of super-duper computer hot-shots out to revolutionize the computer industry. It will be that of competent, high-level computer and system consultants dedicated to improving the tools and procedures of a software development group, at its option, and, in the process, generalizing the improvements and publishing the results (probably jointly with senior and junior Software Group personnel).

4.7 THE PRODUCTS OF THE TECHNOLOGY GROUP

The products of the Technology Group will comprise technical reports, manuals, text-books and presentations, both expository and tutorial, All legitimate media will be employed, including institutional (NASA) reports, proceedings and papers in the professional journals, informal articles in the trade journals, books, and presentations, classes and seminars. As

stated earlier, the essence of the purpose of the group, as viewed by the group itself, is communication: The promulgation throughout the computing community of the results of its own and other people's work (with credit).

The subjects of these communications will lie generally within one of the following topics:

- Industrial standards
- Representation and languages
- Software Production Techniques
- Production Performance Measurement

4.7.1 Industrial Standards

An industrial standard is a criterion of measurement, quality, performance or practice, and may be established in a number of ways. One way, well known in the computer industry, is simply the adoption, by small concerns, of certain of the technical specifications of a line of products of an industrial giant or leader. Other ways include the action of standards committees established by the industry in question, custom, consent or governmental authority. An industrial standard may be technical, in which case it usually specifies what and how. It may be an operative standard, which usually involves human elements, and specifies who, when and why. An industrial standard may also involve both types. Specific examples of standards are:

- Product standards
- Engineering design standards
- Quality standards
- Procedural standards

In considering the adoption of a standard, the maturity of an industry, a product or a production technique in that industry is a factor. Premature adoption of a standard has distinct disadvantages, and failure to adopt at an appropriate time will also have non-trivial drawbacks.

There are a few basic principles with respect to the establishment of standards which the Technology Group will find it well to adhere to. One is that standards are not imposed; they are adopted. The role of the Technology Group with respect to standards should, therefore, not be arbitrary action, but arbitration. The general principles are:

- Standards that are adopted at too early a stage of maturity of an industry, product or procedure are subject to frequent and possibly continual revision in order to keep pace with progress in the corresponding technology. This will defeat the purpose for advancing the standard.

- Standardization tends to inhibit technological progress and development, and to stabilize conditions at the level of development at which it occurs. The implication is double-edged: premature standardization conflicts with orderly development; delayed standardization impairs stability and orderliness once reasonable maturity is achieved.

- The need for flexibility and adaptability to change coupled with the need for, but undesirability of, changing standards implies that standards should be adopted, but that they should be as few in number as is consistent with technological stability.

4.7.2 Representation and Languages

Graphic, verbal and machine languages and conventions are also "standards," but are of a special enough nature and purpose to merit a special category. Of course, software is essentially expressed in terms of various languages; it is not intended that the Technology Group expend any effort in developing new source languages. Examples of the particular kinds of representation and languages to be addressed do include:

- definitions of terms and phrases
- a "requirements" language
- a software structure language
- more formal and useful operational flowcharting conventions
- production scheduling, routing, and assembly forms
- functional and structural specification standards
- representation (symbolic) of hardware/software and software/software interfaces.

Typical examples of words that need explicit definition and universal adoption are correctness, robustness, reliability. A more careful and authoritative analysis should be made of the kinds of errors or bugs that occur in software, so that they can be named and their incidence reported and recorded. Such data will help in developing procedures to reduce errors. The use of jargon may in this way be reduced, and communication between computing personnel in various specialties, installations and parts of the country will be enhanced.

A requirements language is needed to provide for improved communication between user or user representative and software engineers and analysts. The object is to assure that statements of requirements can be set down in explicit and written form by systems or computer analysts and engineers as the result of an operational process analysis and interviews with user personnel. The language should permit representation of procedural steps, data sets, automatic equipment and operator

action and yet be simple enough so that user personnel relatively new to computer and systems work will have little difficulty verifying the written expression of their requirements. Some special form of flow charting, making minimum use of flow charting symbols and virtually no use of highly specialized notation would seem to be appropriate, accompanied by normal text to supplement the flowchart boxes and the describe data sets and sources.

Perhaps the greatest need is for a means of representing a software structure, complete with interconnections (interfaces). The art of representing processes and procedures is highly developed, although further development is required even here. In fact, the representations for structure and process should be complementary; the common element should be descriptions of data sets and structures. This is the greatest deficiency in flowcharting; the emphasis is all on process and sequence. Even here, the data input and output at each step is generally inadequately described, which is responsible for faulty interface design or planning. Intrinsic to the nature of the structural language, in fact, is the ability to represent the connectivity between programs - linkage, parameter passing, data access and identification (for security purposes). This relates, of course, to linkage standards. Once such standards exist, they can easily be graphically represented. Once graphical standards have been adopted, the nesting of computer programs and components to successively lower levels of detail can be meaningfully represented. At that point, experience, intuition, visual perception and the native sense of structural propriety that human beings possess can be fully exercised in developing sound software designs. As in the case of hardware, means of representing structures at all levels will be needed; at the highest level, to provide a preliminary design to accompany general specifications as the basis for user negotiation and contractual arrangement; and at the lower levels, to provide "blue prints" for fabricating software; coding and assembling software components into successively higher levels of subassemblies and assemblies.

Production scheduling, routing, and planning, highly developed production techniques in the hardware world, are at best still in their

infancy in the software world. In this respect, the software industry is still in the age of guilds, in which each component of an end product is hand-crafted with the help of a few apprentices. In today's world of interchangeability, complexity and sheer size, the job must be broken down into layers of buildable and testable pieces, each in turn being an assembly of smaller pieces. Clearly, this requires that the design be appropriate to the end-product's function and operation, but that it also be appropriate for building, assembling and testing. In other words, the procedures used in building and assembling have almost as profound an effect on design at the lower levels as functions do at the higher levels. This statement applies, of course, to both structure and the interconnections of structures at a given level.

There are indications that the nature and function of specifications is not clearly understood by some software specialists. A specification is a design. This is, of course, not the case. A design can exist without a specification, and a specification can, in general, be not fully representative of a design. In fact, a specification is a legal document, an adjunct to a contract, that sets forth a verbal desciption of the item to be purchased. Other adjuncts to the contract include plans, drawings and diagrams to which the specifications may refer. Still other adjuncts have to do with schedules, testing and performance criteria. Thus, work is sorely needed to develop standards for specifications that will complement the representations and languages for requirements, preliminary design, and structure and also consider the nature of the basis of agreement and reciprocal responsibilities between software purchaser and software supplier.

4.7.3 Software Production Techniques

These apply to the processes of design and fabrication, rather than to the techniques or tricks used in programming and coding. Examples are:

- structured programming
- chief programmer teams

- operational assemblies or "builds"
- production engineering

These are relatively new techniques that have been advanced and successfully tested within recent years. They appear to provide good techniques to start with, because they have been successful enough to offer much promise, but yet not so well developed that they can be considered fool-proof.

One of them, structured programming, treats a computer program, system of programs, and program components as structures. This is an approach that merits much more attention and development than it is receiving. The approach highlights the lack of a structural language - that is, a method of representing software structures that is as well developed as the methods of representing hardware structures. Examples of the latter are logic diagrams, wiring diagrams, block diagrams, exploded views, isometric diagrams and so on. These are graphical, but are complemented by the meets and bounds. Therefore, additional initial techniques should include the search for or development of sound structural representations of software and interfaces with hardware and other software.

Another of the foregoing techniques, chief programmer teams, is organizational in approach rather than technological. However, it is a consequence of the structured approach and is therefore closely related to it. If the organizational aspects are stripped from the technique, what is left is an emphasis on the identification and resolution of interface problems as a part of the design process rather than as a part of the debugging process, a much greater emphasis on the importance of the design process as distinct from coding, and the adoption of a certain structural philosophy or software architecture. This architecture is a specific example of the structured approach, and involves the establishment of such structural standards as single program entry points, single program exit points, entries only from and exits to only the next higher program level (no GO TO's), uniform parameter-passing and linkage conventions, etc.

This also points to the need for an effective, unambiguous and standard means of communication on software structure and form between design echelons of the software organization, and between the design and the fabrication groups.

In general, various techniques will apply to various stages of software development. The foregoing two techniques apply primarily to the later stages of design and earlier stages of coding. Techniques applying to statements of requirements, preliminary design and general specifications, and to the system assembly, test, operation and maintenance stages of development and employment should also be sought and modified or developed. These are discussed in various other paragraphs of this section.

Other techniques that may be suitable for the initial stages may be found in the literature (see Section 3). The important thing is that they be compatible with the Software Group's general organization and method of operation. The use of techniques that may have substantial immediate impact on the Software Group should be avoided. It is probable that some "home-grown" techniques can be picked up, modified and improved. This should be done wherever possible.

4.7.4 Performance Measurement

The ability to assess the performance of a software producer, whether an individual or a group, is basic to both the ability to estimate costs and time, and to exercise corresponding control over the production process. At the same time, the ability to measure performance will not of itself provide good production techniques. It will permit responsible management to measure the difference between alternative techniques and procedures. This, of course, is the primary motivation for mentioning performance measurement as one of the more important product categories of the Technology Group.

Performance measurement applies to the management of computer software production rather than to the operation of computer hardware or software. However, the quality of product delivered is certainly an important aspect of performance, and so the measurement of hardware and software performance must be included. However, in this context, performance measurement is the measurement of the capacity of a Software Group to produce working software, together with qualifying measurements of cost, time, and quality of product. This area is closely related to the ability of software management to be able to estimate costs, manhours, manpower skill and level requirements, schedules of software and to exercise some measure of control over its quality.

Basic to the establishment of such measurements is the establishment of some significant portion of the standards, languages, and software production techniques discussed earlier. In fact, the topic of performance measurement is discussed elsewhere in this chapter from other points of view. The point of view here is that science, technology, industry and commerce are based on the ability to measure things or phenomena, describe the results in numerical terms, make comparisons, and make decisions based on these comparisons. For the software industry to emerge as a stable member of the industrial family, its products must be describable and measurable in standardized ways. Predictability of function, performance, cost, size and other qualities then becomes a characteristic of the industry. This is what technology is all about, and assisting in the establishment of sound measurements of performance is perhaps the most important and ultimate job of the Technology Group.

SECTION 5. DATA PROCESSING RESOURCE REPORTING SYSTEM

5.1 INTRODUCTION

At present, computer costs are accurately reported and monitored on Center, leased vs. purchased, size and use category bases, but these costs cannot be related to an actual Project or authorized program. Software costs suffer from the same limitation and in addition some costs are never tagged as software-related. The result is that both development and production costs are impossible to attribute accurately to the programmatic, administrative or institutional end use.

It is estimated that automatic data processing costs represent roughly 10% of the NASA budget. This amount (some $300 million) holds sufficient potential for savings to justify the expense of monitoring expenditures in greater detail, if such monitoring will actually lead to savings. Realizing savings depends on the degree to which inefficient resource use, if it exists, can be revealed by the reporting system and corrected by the monitoring agency. These aspects must be carefully explored before the worth of the system can be forecast.

NASA should establish a reporting system for data processing resource expenditures. This would record the application of resources -- manhours, machine time, supplies and other operating expenses -- to each project or administrative category. Periodic processing would report detailed information locally and summary data to Headquarters for measuring expenses against budgets, comparing overhead rates among locations, accumulating the cost of software produced, etc. A software inventory, keyed to project identifications, would store descriptions of the programs produced or worked on. The inventory, started from scratch at inception of the reporting system, would build up in time to provide a basis for exchanging programs (or preventing the duplication of existing ones).

5.2 BENEFIT AREAS AND POSSIBLE UTILIZATION

A reporting system of the kind proposed is basic to managing activities as diverse and complex as NASA automatic data processing. It would enable better budget planning, at Center level as well as Headquarters, through its record of activities in past budget periods. It would make possible a reading on all ongoing software activities, which could help get together NASA personnel interested in the same application. As explained above, it would also produce a software inventory.

By assigning costs to the activities which ultimately benefit from them, the monitoring system could make possible a more precise determination of priorities for additional resources. More accurate reporting of resource use should also enable computer center managers to spot imminent trouble spots, e.g., excessive debug time on a new program. It would also make it possible to compare activities at similar centers, possibly revealing surpluses or deficiencies in computing capacity or programming skills.

The writers have been made aware of both Congressional and Executive Branch expressions of concern over the need for more precise control measures to manage ADP in the Federal government. A system like

the one proposed is a first step toward meeting these requirements, which are becoming increasingly stringent, for management reporting to external budgetary or audit agencies.

5.3 SCOPE

A concept would be developed for a standard resource expenditure monitoring and review system for use throughout NASA. A software inventory system would be an adjunct to the monitoring system.

System objectives and information requirements would be developed with NASA Headquarters assistance. A limited system description based on samples of field practice would then be assembled. Subject to favorable NASA review, the description would supply the basis for subsequent system definition, design and implementation.

5.4 SYSTEM DESCRIPTION

A standard reporting and monitoring system of the kind proposed includes a system description, operational procedures and supporting computer programs and their documentation. The procedures specify what actions are to be taken, where and by whom, and set standards for reporting media. The computer programs accomplish necessary processing of reporting and monitoring data. Program documentation describes the programs sufficiently to enable their use and revision by programmers. It also specifies input and output formats and media, which ensures the effective interchange of information throughout the system.

Reports in such a system fall into two categories: expenditures during the reporting period and inventory at its conclusion. Current hardware and software modules, complete or under development, comprise the inventory.

"Expenditures" include the cost of equipment and supplies, payrolls, facility operation expenses, maintenance and software contracts,

etc. A report will show the distribution of these costs among products and overhead of the facility reporting them.

"Products" may be either developmental or production; that is, a computer facility may produce programs as an end product, or it may do production computing. Activities which don't contribute directly to a recognized product are necessarily overhead. By recognized product is meant an identified part of some approved project or program.

Overhead costs are charged to each product in proportion to the amount of direct resources applied to the product. If in a given month a product requires 1% of a computing facility's available direct resources, then it acquires a 1% share of that facility's overhead for the month in question. This is necessary to show the true cost of products, without charging them for more than a fair share of overhead.

The management philosophy such a reporting system might support is as follows: First, one makes sure all expenditures are assigned either against an authorized product or overhead. Next, one compares facilities for overhead which is too high; these are subject to remedial investigation. Finally, one checks that products are not being over-charged for computing services; if none is, then computing resources are being used with a fair degree of efficiency.

There are a number of areas of potential difficulty which face a system implementation of the sort being considered. An obvious one is the diversity of machine types in NASA on which programs comprising the system would have to run. Then there is the question of how to treat Category B computers in the reporting system. These machines will obviously not all have data recording programs, and those which do exist may not supply sufficient input data for the standard resource reporting system. The expense of creating data collection programs for these computers may justify, as an alternative, the use of simplified procedures. For example, it may be possible to record use data manually for these computers in a log maintained by the operator, if utilization is relatively light.

Complications may occur in reporting design and programming effort. While we can conceive of in-house programmers' filling out of time cards, how will the programming efforts of a NASA scientist be reported? Of a staff member of an academic institution under NASA contract? Professionals who program occasionally should be exempt from such reporting, but would contractors' programming staffs be similarly exempt? Could compliance with reporting requirements be made relatively inexpensive for NASA contractors?

These questions of how broadly to implement the reporting system, as well as ones concerning the uses to which it can beneficially be put, will be answered in the development of a system concept. This effort will demonstrate the feasibility and worth to NASA of a teleprocessing resource reporting system.

SECTION 6. A NASA GENERAL USE COMPUTER NETWORK

6.1 INTRODUCTION

Data communication is an integral part of NASA mission activities. The amount of data flow via communication carriers is expected to increase by several orders of magniture by 1980. The amount of this data, and the numbers and geographical dispersion of experimenters requiring access to it for processing and analysis, suggest additional data communications requirements of large proportions.

There is a trend in NASA toward remote access computing, both inter- and intra-center. Existing general purpose centers are beginning to provide more and more remote job entry and interactive time-sharing services to their users. Industry experts predict that by the 1980's, the majority of general purpose computer usage will be accessed remotely.

NASA leases its communications services from regulated carriers. The recent rise in competition for carriers of data communications and the tariffs these competitors propose for such service suggest that end

users may experience significant cost reductions as a result. In addition, entirely different services, much more useful for interconnecting computing systems, are under development, and petitions to furnish them to the public have been made to the Federal Communications Commission. Anticipating the availability of improved data services, NASA requested the writers to evaluate the impact of these developments in the domestic data communications field.

6.2 DATA COMMUNICATIONS' IMPACT ON NASA COMPUTING

The new capabilities, as presently planned, would facilitate computer networking, itself in early development stages. This suggests the development of NASA standards for intercommunication languages and protocols, in order to promote the use of ubiquitous commercial data services forecast in the 1980's. This section describes a project to perform some preliminary forecasting and planning necessary to the development of the standards.

A necessary first step is a forecast of data communications services available in the 1980's and verification of the impact on NASA computing. Then, the plans of government and industry standards groups should be investigated, NASA requirements for intercomputer communications should be ascertained, and the unsolved technical problems of computer networking should be identified. The result will be used to plan for developing the necessary standards.

6.3 BENEFIT AREAS AND POSSIBLE UTILIZATION

The planned commercial data communications capabilities, as currently understood, will mitigate the problems of using present analog communications for intercomputer data exchange. This would leave the user of the new facilities free to concentrate on interchanging information among computers and persons in a geographically dispersed network. Among the advantages of a network are: (a) the avoidance of duplicated

data bases, (b) specialization of facilities and associated software, with increased efficiency, (c) the convenience of remote computation and its corollary, instantaneously available information however remotely located, (d) synergisms growing out of associations among programmers or others, made possible by the network, (e) the sharing of programs and general purpose computing facilities, and (f) the use of inexpensive miniature computers as intermediaries between people and large computing complexes.

Networking is in its early development and not yet capable of delivering the cited advantages. Except in systems designed as a network from their inception, the exchange of information among machines is prevented by their many differences in function, data and operating programs. Short of eliminating all differences between NASA computing systems, the only means of transferring information readily is standardizing the various interfaces between people, data, computing equipment, programs and communications channels.

It is appropriate to establish the goal of a computer network which is as transparent to data processing resources as the present voice network is to people. The effort required to determine whether this goal is fully attainable will do two things: establish the boundaries of what is possible and desirable in the way of computer networking, and develop techniques required for successful networks having less ambitious goals than that of total generality.

6.4 SCOPE

A study would be made to identify actions necessary to achieve general NASA computer networks based on commercial data services available after 1975. These actions would include determination of the needs for computer networks in NASA and development of the following:

(1) A design for a computer network for general NASA use. Design restrictions on computer systems desiring to use the network would be limited to observing standard forms for exchanging data and programs and for invoking processes remotely

(2) Computer network components for general NASA use in implementing standard interfaces

(3) Standards for the interface among NASA computer systems, which includes program-program, program-data, system-communications, and system-system interfaces

(4) A prototype computer network.

A preliminary effort should be made to complete the following actions:

(1) Verification of the plans of DATRAN, Bell, Western Union, et al., for future domestic data services offerings.

(2) Assessment of the progress and plans of commercial and government computer network developments, such as the ARPA and Octopus networks, TYMNET, etc., to verify the suitability of the proposed NASA network

(3) Determining the applicability of planned and existing NASA, government and industry standards

(4) Identification of the unsolved technical problems of networking

(5) Survey of NASA networking requirements

(6) Determination of initial requirements for networking standards.

The results would identify the sort of network NASA could establish and the steps required to do so.

6.5 DEFINITIONS

Sharing of computing equipment means use of a computer by a remote entity, or borrower, such as another computer or a person

interacting via a terminal. Programs and data local to, and designed to run on, the shared computer may be used, in which case they too are shared. The remote user must know how to address the intended host processor, invoke the program and interpret the results. This knowledge is broadcast, in the current state of the art, in the form of computer-based directories and, for human users, printed manuals. If the borrower's programs, remote from the shared computer are used, they may require adaptation to the intended host computer. This can involve a different language or dialect, a different operating system, different machine language, and device differences (e.g., word length). In general these differences are overcome, if at all, by painstaking human labor, which is required to recode programs. For this reason, such processing of foreign programs is not done in current networks, save those which are highly constrained.

Network sharing of programs, meaning their use by remote as well as local processors or persons, involves either running the programs on their "home" processor or transferring them to a foreign host; both cases have just been discussed, above.

Sharing of data (made accessible to remote users) requires its structure and format to be communicated to some computer program for processing the data. This may be accomplished by publicizing the data's content and structure, as well as protocols necessary to obtain or replace it through intermediate or custodial processors. This technique breaks down, if there are many different protocols and data structures, because of the complexities of accommodating them all. The sharing of storage facilities requires the same techniques except that the data's content, being private to the "owner" of the data, is not publicized.

6.6 COMPONENTS OF A NETWORK

Besides communications and computing systems, a data processing network includes components to govern its operation, of which the following is a brief list:

Executive Program. The executive assigns workloads and resources, controls restarts, maintains directories, administers priorities and the graceful degradation of capabilities, etc. It may constitute an added part to the operating systems of the network's constituent computers, or it may occupy one or more computers totally.

Directories. These list subscribers and addresses, programs, data, computing systems, languages, etc. available within the network. Complete descriptions of the listed entities are a part of the directories and serve as the basis for accessing these resources.

Inter-system Language and Interpreter. By means of this language network users and programs communicate with each other and with the executive, in order to invoke processing remotely, acquire data, etc. The interpreter transforms requests couched in this language to calls for resident procedures, which perform the required processing.

Communication Channel Protocols. The logical interface with the communications system, consisting of control signals necessary to obtain and terminate service and indicate an addressee, may not be subject to NASA standardization. Error detection and correction techniques, bit and framing patterns, multiplexing, synchronization procedures, etc., are.

Standard Data Interface. This consists of a data description language and interpreter by means of which general data types and structures can be described and hence processed. This component in effect removes obstacles to the ready exchange of data throughout the network.

Programming Languages and Compilers. These components are the least susceptible to standardization owing to the loss of generality incurred thereby. The many specialized dialects apparently give a richness to these languages whose loss might not be offset by the advantages of standardization. In addition, the many differences in commercially supplied processors and operating systems are beyond the

immediate ability of NASA to resolve. It may be preferable to abandon the objective of transferring programs to processors in the network without restriction.

6.7 DISCUSSION

The most significant aspect of the new data communication services appear at this time to be the boost they would give to exchanging information among geographically distributed people and computing equipment -- in a word, to networking. In the words of one Datran executive, their data network will be "transparent" to the user and will perform all communications operating functions, such as circuit selection and switching, technical control (detecting and removing poor quality circuits from service), routing error control, etc. The Datran system will furnish switched circuits, which avoid many of the opportunities for error found in a store and forward message processing system. As in the Bell voice system, only a very small repertoire of commands is required of the user, consisting of requesting service, designating the addressee, disconnecting, etc. Unlike the voice system, the circuit will be supplied within milliseconds, be very quiet, and not be subject to the momentary interruptions, common in the voice network, caused by rerouting.

These circuits will not be unlike data channels internal to most computer systems, differing only in their higher error probability, the possible need for privacy stratagems and the fact that they can serve as conduits for foreign (and hence possibly unintelligible) information. The first two aspects present little difficulty, being subject to control by well-understood techniques. Translating between local and foreign data formats and program languages is also readily accomplished, but its implementation tends to exceed total system resources, without severe limitation on the numbers of languages and formats the system must successfully interpret.

The chief obstacle to computer networking in NASA is the diversity of equipment, procedures, codes, etc. in use. Designers of networks of computers have dealt with this problem in either of two ways. The first requires a rigid specification of all the information which may flow in the system, and is employed in networks each of whose components was especially designed (or modified) to function as part of the system. The second permits rejecting or ignoring information in parts of the system where it is unintelligible, and is adopted for assemblages of computer systems interconnected for the convenience of their human users, who supply most of the knowledge of protocol needed for remote access.

Either of these approaches limits the scope of the network. In the first case, the programming effort required to adapt software and hardware prevents the use of more than a few different computer systems. In the second case, component systems are incapable of using the network without human initiative and intervention. People, in turn, are limited in the amount of protocol and number of programming languages they can remember or use. With these limitations, attainment of the advantages of networks is similarly limited.

The advantage of having networking standards is that their existence encourages building compatability into data systems when they are developed, at great savings. Adding communications interfaces to existing systems can be prohibitively expensive and a penalty NASA can avoid by developing standards for teleprocessing networks.

SECTION 7. CONFIGURATIONS FOR PROCESSING MISSION CONTROL DATA AND TELEMETERED EXPERIMENT DATA

7.1 INTRODUCTION

There are a number of trends in the processing of mission-related data throughout NASA. One of these is the growth in the amount of data expected from space experiments, particularly those in the ERTS and GARP programs. The increased data flow will require storage and processing facilities much larger than have previously been installed by NASA. The change in magnitude is great enough to require a new design for the facilities, rather than a scaling-up of current plant.

A second trend, arising from the changing character of experiments, is toward greater control over satellite platforms by experimentors. Experimental investigators now participate in controlling the satellite vehicle, re-directing its course or adjusting its sensors to acquire more valuable data. This means that experimental data must be processed much faster than before to serve as a basis for readjusting the process by which it is acquired. New projects requiring this capability are being supplied with their own data

processing facilities, to guarantee the timeliness of data reduction and analysis processing. This is a departure from current practice, namely, employing a data reduction and storage center shared by all space experiments, and it raises the question of optimality, because central data reduction and storage facilities are duplicated, on a smaller scale, for each such project.

Another aspect of the changing nature of experiments is the emergence of multi-disciplinary teams to replace independent investigators as the dominant mode of NASA experiment design and direction. The cooperation among users of experimental data implied by this development may make possible their sharing of processing and software resources. This could achieve economies over the present practice, which is to leave the analysis programming and processing to numerous independent Principal Investigators and whatever arrangements they can make.

Another trend is a migration in the locus of space experimentation to include the manned space flight centers at Huntsville, Alabama and Houston, Texas and a reverse migration, in the locus of mission control activities, to Goddard Space Flight Center. Several developments, taken in the context of this migration, argue for viewing the geographically-dispersed resources involved as a system. The first of these is the prospect of joint endeavors by two or more NASA centers as, for example, in the GARP and the Shuttle program. A second is the increasing prevalence of project-unique control centers, which require high reliability; this is achieved by redundancy, which in combination with intervals of inactivity between missions is attended by low utilization. A third development, the Tracking and Data Relay Satellite, will funnel all data from space into a single earth station, strongly suggesting bulk reduction at this location to save on costs for its subsequent distribution to users.

These developments represent an opportunity to redistribute computation processes, software development and storage of data at a net savings, because of any unused computing capacity of redundant equipment, and economies of scale in storing data and developing software. Growing

commercial data transmission capabilities will provide services to make this approach economically feasible.

An effort should be made now to determine optimum configurations of computing resources for reducing and processing telemetered data in the 1975-85 decade. The object would be to apply technical improvements - now emerging - to relevant data processing resources viewed as a system, rather than piecemeal. Hence, processing functions which support mission operations as well as space experiment data handling would be examined to determine their best geographical distribution. Alternative configurations (e.g., project-unique vs. NASA Center-unique analysis facilities) of computer-related resources and their intercommunication would be evaluated in terms of cost, reliability, responsiveness and other performance characteristics required by NASA.

7.2 SCOPE

The following data handling functions are the subject matter:

- Processing experiment data from space and aerodynamic vehicles
  - Input processing, reduction and storage
  - Analysis
  - Command generation

- Supporting space operations
  - Mission planning
  - Vehicle trajectory monitoring
  - Biomedical and vehicle systems monitoring
  - Vehicle systems operation.

The entire effort, resulting in a set of operational data systems, spans three phases: feasibility, system definition and design/implementation.

NASA centers under OSSA and OMSF to be included in the feasibility study are Goddard SFC, Marshall SFC and MSC. An analysis of processing requirements will concentrate on AE, HEOS, ERTS, Skylab, GARP and Shuttle programs, while a review of present and planned facilities will center on TELOPS, TDRS and project control centers at GSFC, MSFC and MSC.

7.3 POSSIBLE BENEFITS

At present, the development of programs for analyzing space experiment data is not centrally managed, being under the supervision of the numerous individual Principal Investigators. The possibilities for duplicate programming owing to this situation could be reduced if a joint-use center were designated for analysis purposes, permitting a concentration of programmers and their management at the center. Production by these programmers could be improved not only by avoiding duplicate program development, but also by providing the programmers complex software to assist their efforts.

Analysis processing would be invoked at these centers by possibly distant experimenters through terminals, telecommunications and interactive time-sharing software. These could also be used to transmit the experimenter's data to him on-line, avoiding part of the three and one-half month delay involved in the current off-line process. An advantage is an expected reduction in amount of data requested, if experimenters are able to browse through it prior to making their requests. Further reductions are anticipated as a result of the rapidity with which data can be received by the requestor, which may tend to forestall requests for exhaustive data sets.

If there are a number of analysis processing centers, each may specialize in some aspect of processing, relieving others of any workload in its area of specialty. This eliminates duplicative software development and maintenance efforts by assigning them uniquely. Communications, linking a network of centers and terminals, would provide user access to

any center for specialized computing, or for backup in case of a failure at one of the centers.

An important benefit of establishing networks of computing centers is the opportunity this offers to standardize features. This can lead to the ready exchange of data and programs conforming to standard, which eventually will be plentiful, owing to the widespread use an acceptable standard may attract solely by its existence.

There may also be potential for establishing a shared mission control processing center or network of centers, in order to balance out the cyclic inactivities occurring in individual projects. An integration of the processing support to several projects in this way may make it possible to share redundant equipment, needed for reliability, with a consequent reduction in costs. The greater savings, however, are due to the use of common software.

7.4 CONCLUSION

Two sometimes conflicting goals motivate the search for the best distribution of data processing resources and functions. The first of these is assigning control over the resources to the man responsible for their ultimate product. The second goal is the greatest economy consistent with meeting performance objectives. The goals conflict in the case of the small user who for economy's sake must use a computer service, over which he has no control. To assign control in such cases usually means assigning excess resources, because of their indivisibility within relatively large increments of capacity. He regains a measure of control if there is a competing service to which he can transfer should the original service prove unsatisfactory.

This study should examine the technically feasible configurations of resources for telemetered data and mission control functions and evaluate their contributions toward these goals. If low, new or refined feasible configurations would be sought and evaluated until their contributions are acceptably high.

SECTION 8. AN INTEGRATED BUSINESS DATA PROCESSING SYSTEM

8.1 INTRODUCTION

In commercial practice, business data processing is closely coupled to operations, and its proper functioning can be critical to financial success. Hence it is accorded a high priority on the use of resources. In NASA, however, business data processing is at the end of a long list of mission-related ADP functions and in consequence may experience difficulty in obtaining qualified personnel and technically advanced equipment and software.

A deficiency of either personnel or computing components is undesirable because it inevitably causes deterioration in service. It may also be self-perpetuating, because less advanced operations tend to attract less qualified personnel, who in turn may be less skillful in explicating the need for improvements and fail to "justify" the required fuding.

Continuing government pressure for control of Federal expenditures is an established trend which can be expected to continue for the next few years at least. It is one which would heavily burden current Federal

Government business data systems. Demands will be for information which is more current and possibly more detailed or precise than that available today. At the same time, managers may require more freedom in structuring retrieval processes to help them gauge the impact of contingencies, like a shift in funding emphasis. Effective ADP support to these activities will require a contemporary system design employing a carefully structured data base and interactive software to achieve the rapid response required. Systems of this sort do not form a part of NASA business data systems at present.

The trend toward increased scrutiny of Federal expenses carries over to ADP resource management, too. Here, the need for economy and a desire for uniform reporting argue for greater integration within and between similar government systems, such as those for personnel management data. There is a lack of coordination among the many NASA Center production processes, files and data which may be similar, such as the various Center payroll systems. Some of these systems, which may be under continuous revision, could be standardized among Centers with savings in development and maintenance costs. Perhaps the responsibility for each could be assigned to a single, possibly different center.

NASA should determine the feasibility and advantages of integrating portions of the management information files, programs and data existing at or planned by the various NASA Centers and Headquarters. The objective would be to avert unnecessary duplication by sharing programs, data, possibly computation power and know-how among the Centers and Headquarters. By integration is meant design revision to exploit latent commonalities among data, processes or outputs for contributions to efficiency.

8.2 SCOPE

Business data processing as used here means the support given to administrative and management functions in NASA. It includes financial, planning, and commodity or project management data and programs. Taken

together these files represent the machinable component of information used to manage NASA; hence the term management information system may be used.

The advantages and disadvantages of integrating the ADP support to management in NASA should be developed in detail. Opportunities for integration between and among Centers should be examined first, with intra-Center integration following as a consequence.

The outcome would be a list of the NASA business data processing functions which could feasibly undergo some degree of integration, reasons why they should (or shouldn't) and description of the information system components which would result. When approved by NASA, this would constitute guidance to commence detailed definition of an integrated management information system. A plan for such a system definition could then be produced.

8.3 POTENTIAL BENEFITS

The benefits of integrating a system arise from the design process of organizing all resources for achieving system objectives. Benefits consist of faster access to more data by accredited individuals, such as managers; a minimization of unnecessary duplication of effort; and uniformity among operations, facilitating their comparison by management and reducing training and maintenance requirements. A sometimes beneficial side effect, incidental to the definition of system objectives, is the illumination of organization goals and procedures, which may as a result emerge in more explicit -- and internally consistent -- form.

If general purpose file management software is part of the integrated system, it further reduces development, maintenance and training costs in proportion to the breadth of its application. Such software is called a data management system (DMS).

The integrated system should be readily accessible to NASA persons needing the data it contains. These persons are not likely to

have computer programming experience and a means for them to communicate with the system in relatively natural fashion is desirable. For this, a procedures language couched in terms familiar to users of the system is necessary. When given on-line access to a system via a terminal and such a language users may readily acquire proficiency in its use. The result can be very widespread use of the integrated system by NASA, with attendant benefits derived from the immediacy with which current data is available.

An integrated system will facilitate the interchange of data necessary to the painstaking management required in the current era, with its emphasis on efficiency. Uniform data, reported in uniform ways, will enable effective management review of operations throughout the NASA organization.